XOR Convolution and the Fast Walsh–Hadamard Transform

The Fast Walsh–Hadamard transform is one of the cleanest examples of how a structured change of basis can transform an expensive combinatorial operation into a simple elementwise multiplication.

Most people encounter the FFT through ordinary convolution:

- transform,
- multiply,
- inverse transform.

The Walsh–Hadamard transform does essentially the same thing, but for a different geometry:

- the geometry of bit flips,
- XOR translations,
- and Boolean hypercubes.

This gives rise to XOR convolution, also known as dyadic convolution.

---

1. The Hadamard matrix

The basic Hadamard matrix is:

The basic Hadamard matrix is:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Larger matrices are constructed recursively:

$$H_{2n} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}$$

For example:

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

After normalization by $1/\sqrt{n}$, the transform becomes orthogonal:

- preserving lengths,
- preserving inner products,
- acting as a rotation/reflection in high-dimensional space.

Unlike Fourier transforms, the Walsh–Hadamard transform uses only:

- additions,
- subtractions,
- sign flips.

No complex arithmetic is required.

---

2. Construction by index parity

A particularly elegant way to define the Hadamard matrix is through bit parity.

If row and column indices are represented as binary vectors:

$$a, b \in (\mathbb{Z}_2)^m$$

then the matrix entries are:

$H_{a,b} = (-1)^{a \cdot b}$

where:

- $a \cdot b$ is the bitwise dot product modulo 2,
- the result is either $+1$ or -1.

For example:

a=101, b=110

gives:

$a \cdot b = 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 0 = 1 \pmod 2$

so:

$H_{a,b} = -1$

This parity construction is one of the deepest aspects of the transform.
The basis functions are parity functions over subsets of bits.

---

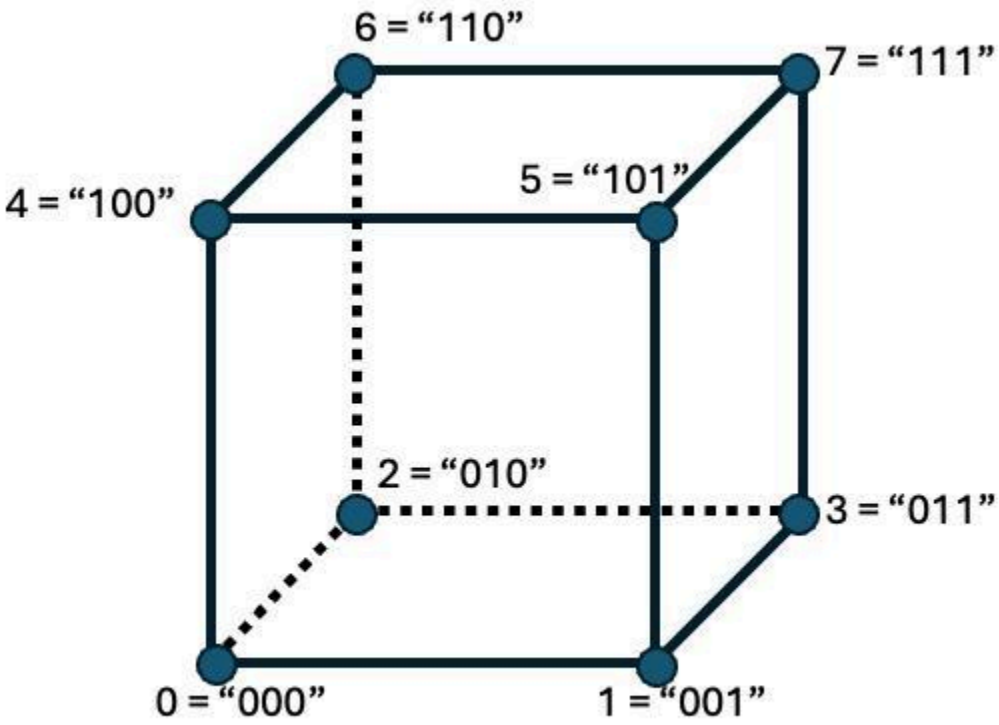
3. The Boolean hypercube

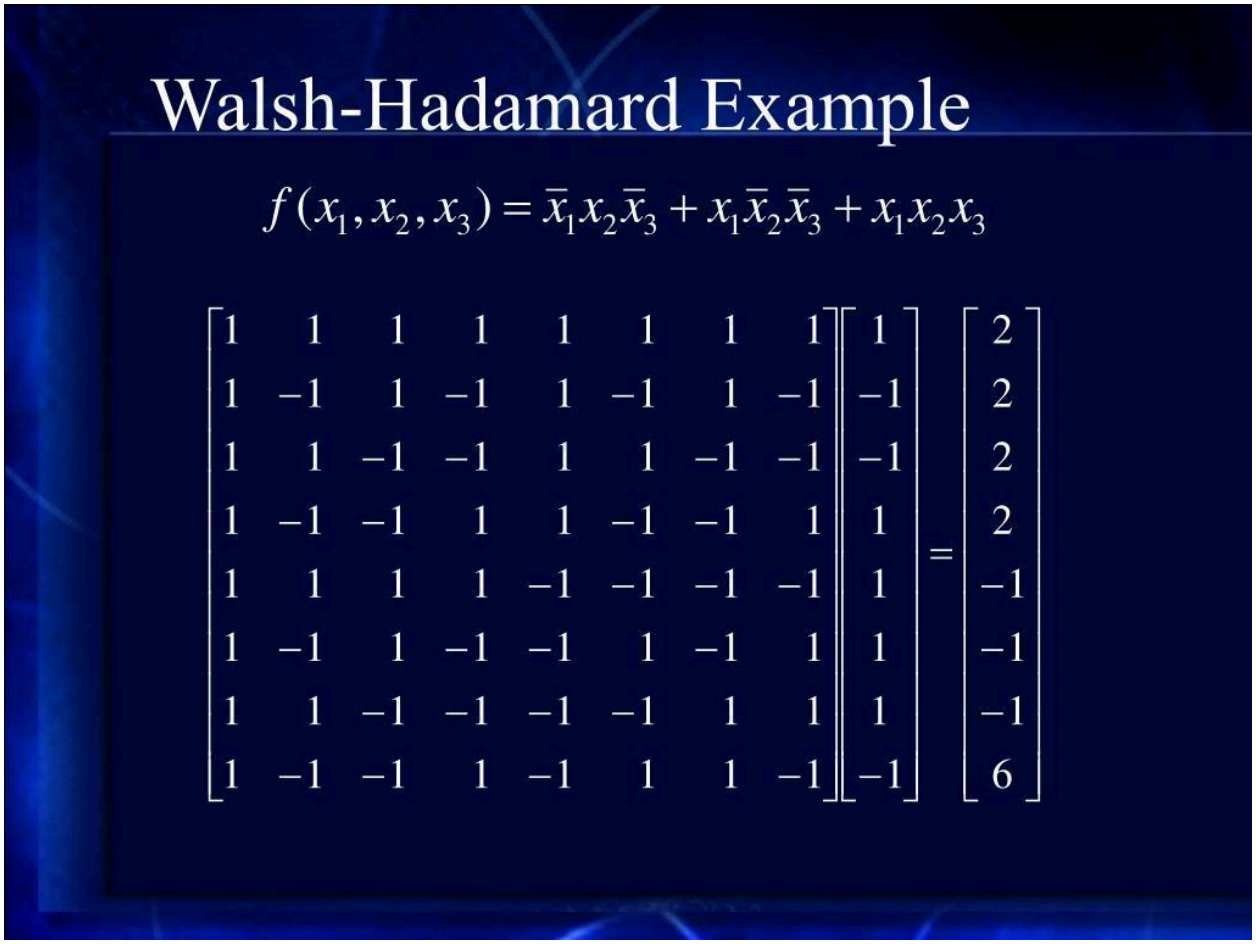
Indices of length m correspond naturally to vertices of an m-dimensional Boolean hypercube.

For 3 bits:

000, 001, 010,..., 111

form the 8 cube vertices.





Entry 5 (101), 6 (110) is -1 by the parity calculation (row,column,zero indexing). Or you can swap rows and columns. The rows/columns are the Walsh basis functions.

The Walsh basis functions are parity-wave patterns over a hypercube.

This leads to a useful intuition:

Fourier analysis	**Walsh–Hadamard analysis**
waves on circles	parity waves on hypercubes
translation by addition	translation by XOR
sinusoids	parity sign patterns
frequency modes	parity modes

For this reason, the Walsh–Hadamard transform is often described as:

the Fourier transform of the Boolean cube.

---

4. XOR convolution

Ordinary convolution uses addition/subtraction of indices.

XOR convolution instead uses bitwise XOR:

$$(f *_{\oplus} g)(k) = \sum_i f(i)g(i \oplus k)$$

Geometrically:

- ordinary convolution slides signals through Euclidean space,
- XOR convolution flips signals through hypercube dimensions.

If:

k=1010

then:

- bit 1 is flipped,
- bit 3 is flipped.

So XOR convolution measures overlap under structured bit-flip symmetries.

---

5. Why the transform works

The Walsh parity function is:

$$\chi_a(x) = (-1)^{a \cdot x}$$

where:

- a and x are binary vectors,
- $a \cdot x$ is the bitwise dot product mod 2.

The output is always:

- $+1$,
- or -1.

So $\chi_a(x)$ is really a parity detector.

The Walsh basis functions satisfy:

$$\chi_a(x \oplus y) = \chi_a(x)\chi_a(y)$$

This is directly analogous to:

$$e^{i\omega(x+y)} = e^{i\omega x} e^{i\omega y}$$

in ordinary Fourier analysis.

As a result, XOR convolution diagonalizes in Hadamard space:

$$\widehat{f *_{\oplus} g} = \hat{f} \odot \hat{g}$$

So the algorithm becomes:

1. FWHT of f
2. FWHT of g
3. elementwise multiply
4. inverse FWHT

Complexity drops from:

$O(n^2)$

to:

$O(n \log n)$

---

6. The butterfly structure

The Fast Walsh–Hadamard Transform uses the recursive operation:

$(u,v) \rightarrow (u+v,u-v)$

At each stage:

- pairs differing in one bit interact,
- information mixes across one hypercube dimension.

After $\log_2 n$ stages:

- every output depends on every input,
- through structured parity interactions.

This creates:

- dense global mixing,
- while avoiding dense matrix multiplication costs.

---

7. Multiplicative gating interpretation

One especially interesting aspect of XOR convolution is its multiplicative structure in transform space.

Each transformed coordinate behaves independently:

$$\hat{h}_i = \hat{f}_i \hat{g}_i$$

This acts like a distributed multiplicative gating system:

- if one component is near zero, the interaction vanishes,
- if both are large, the interaction becomes strong,
- signs determine reinforcement or cancellation.

This gives XOR convolution properties resembling:

- coincidence detection,
- feature agreement,
- spectral masking,
- parity-mode resonance.

Because Walsh coordinates correspond to global parity patterns, the interactions are:

- distributed,
- combinatorial,
- highly nonlocal.

---

8. Harmonic eigenmodes of XOR geometry

The Walsh basis vectors are the natural harmonic modes of XOR translation.

This is the Boolean analogue of Fourier frequencies.

A Walsh mode is preserved under XOR shifts up to a sign:

$$\chi_a(x \oplus k) = \chi_a(x)\chi_a(k)$$

This means Walsh functions are eigenfunctions of XOR translation operators.

Geometrically:

- ordinary Fourier modes describe oscillations on circles or Euclidean domains,
- Walsh modes describe parity oscillations on hypercubes.

This viewpoint helps unify:

- coding theory,
- Boolean analysis,
- cryptography,

- hyperdimensional computing,
- associative memory,
- randomized transforms,
- neural mixing architectures.

---

9. One-hot versus Walsh inputs

Two complementary viewpoints are useful.

One-hot inputs

A one-hot vector corresponds to:

- a localized hypercube vertex.

Interactions become dependent on:

- XOR displacement,
- relative bit flips.

Walsh inputs

A Walsh basis vector corresponds to:

- a pure parity harmonic.

Interactions become:

- spectral correlations,
- parity-mode filtering.

This duality is closely analogous to:

- impulses versus frequencies in Fourier analysis.

---

10. Why it matters computationally

The FWHT provides:

- global orthogonal mixing,
- structured high-order interaction,
- O(nlogn) complexity,
- simple hardware-friendly operations.

This makes it attractive in areas such as:

- compressed neural mixing layers,
- associative memories,
- fast randomized projections,
- binary embedding systems,
- error-correcting codes,
- hyperdimensional computing.

In many practical systems, it provides a useful middle ground between:

- purely local operations,
- and fully dense matrix multiplication.

---

11. Summary

The Walsh–Hadamard transform can be viewed as:

- a Fourier transform over the Boolean cube,
- a decomposition into parity harmonics,
- and a fast structured mixing operator.

XOR convolution then becomes:

- convolution under bit-flip geometry,
- diagonalized by parity-wave eigenmodes.

The result is a surprisingly rich computational framework built from:

- sign flips,
- additions,
- parity structure,
- and hypercube geometry.